FREE eBook

LEARNING flexbox

Free unaffiliated eBook created from
Stack Overflow contributors.

#flexbox

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: flexbox

It is an unofficial and free flexbox ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official flexbox.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with flexbox

Remarks

This section provides an overview of what flexbox is, and why a developer might want to use it.

It should also mention any large subjects within flexbox, and link out to the related topics. Since the Documentation for flexbox is new, you may need to create initial versions of those related topics.

Versions

Version	Release Date
Candidate Recommendation March 2016	2016-03-01
Candidate Recommendation Sept. 2012	2012-09-18
Working Draft March 2011	2011-03-22
Working Draft July 2009	2009-07-23

Examples

Installation or Setup

Flexbox is a CSS3 module, standardized by the World Wide Web Consortium. It is a layout mode for element arrangement such that the elements behave predictably when the page layout must accommodate different display sizes.

Because it is a part of CSS3, you don't need to install anything. It can be used as long as the browser supports it, and most modern browsers do. To check if your browser supports it or not, here is the compatibility chart.

To set up and use `flexbox` in your CSS, simply add `display: flex` to a selector.

Flexbox Containers and Items

Flexbox or flexible box is a layout method for arranging content on a page in a predictable manner. Flexbox provides an improvement over traditional block model positioning using floats or even table like positioning for content on the page.

At its core, Flexbox can be broken down into a parent element (flex container) and a child element (flex item).

Flex Container

A flex container can be created by setting its display property to flex:

```
.container {
  display: flex;
}
```

Flex item

Every child element of a flex container becomes a flex item. These flex items can then receive additional properties to modify how its positioned on the page.

```
.item {
  flex: 1;
}
```

This `flex: 1` property is shorthand for `flex-grow: 1` enabling it to grow relative to its siblings within the container.

Putting these together this is the HTML markup:

```
<div class="container">
  <div class="item"></div>
  <div class="item"></div>
  <div class="item"></div>
</div>
```

Flexbox Introduction

The Flexbox (Flexible box) layout, introduced in CSS3 provides a more efficient way to lay out, align and distribute space among children elements(`flex items`) within a container element(`flex container`). Most importantly even when their sizes are unknown or dynamic and hence the term "flex" or "flexible".

Lets start with the basics and see how an container can be initialized as flex.

Consider the following markup:

```
<div class="flex-container">
  <div class="flex-item-1"></div>
  <div class="flex-item-2"></div>
  <div class="flex-item-3"></div>
</div>
```

A flexbox is initialized by simply using `display: flex`. Done!

```
.flex-container {
  height: 100px;
  padding: 10px;
  background: grey;
  display: flex; // or display: inline-flex;
}
```

Now that the flex container is ready, let us play around with its flex items giving them a width of say 25% each and horizontally center aligning the flex items inside their parent container.

```
.flex-item-1 {
  width: 25%;
  background: green;
}

.flex-item-2 {
  width: 25%;
  background: purple;
}

.flex-item-3 {
  width: 25%;
  background: pink;
}
```

Please note, I have purposely given the 3 children a width of 25% each to show how easy it is to move around the children within a flexbox.

Now when we run the above code, we should see the 3 children within the container horizontally next to each other. This is because by default, a flexbox has the property `flex-direction: row`. Meaning by default the flex items will be aligned horizontally next to each other. Also there should be a gap on the right since the total width of the children didn't add upto 100%.

Now try adding the property `justify-content: center` to the `flex-container`.

```
.flex-container {
  ...
  justify-content: center;
}
```

The result would be the children being horizontally center aligned. The same property has other values such as `flex-end`(align to the right), `space-around`(equal space around the items), `space-between`(equal space between the items).

> *Important: Other block element properties like `text-align: center` etc. has no effect on a flex element.*

Read Getting started with flexbox online: https://riptutorial.com/flexbox/topic/8925/getting-started-with-flexbox

Credits

S. No	Chapters	Contributors
1	Getting started with flexbox	Aaron Vanston, Cassidy Williams, Community, Nikhil Nanjappa